MANIFOLD ARRAY PROCESSOR

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to processing systems in general and, more specifically, to parallel processing architectures.

Description of the Related Art

Many computing tasks can be developed that operate in parallel on data. The efficiency of the parallel processor depends upon the parallel processor's architecture, the coded algorithms, and the placement of data in the parallel elements. For example, image processing, pattern recognition, and computer graphics are all applications which operate on data that is naturally arranged in two- or three-dimensional grids. The data may represent a wide variety of signals, such as audio, video, SONAR or RADAR signals, by way of example. Because operations such as discrete cosine transforms (DCT), inverse discrete cosine transforms (IDCT), convolutions, and the like which are commonly performed on such data may be performed upon different grid segments simultaneously, multiprocessor array systems have been developed which, by allowing more than one processor to work on the task at one time, may significantly accelerate such operations. Parallel processing is the subject of a large number patents including U.S. Patent Nos. 5,065,339; 5,146,543; 5,146,420; 5,148,515; 5,546,336; 5,542,026; 5,612,908 and 5,577,262; European Published Application Nos. 0,726,529 and 0,726,532 which are hereby incorporated by reference.

One conventional approach to parallel processing architectures is the nearest neighbor mesh connected computer, which is discussed in R. Cypher and J.L.C. Sanz, SIMD Architectures and Algorithms for Image Processing and Computer Vision, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 37, No. 12, pp. 2158-2174, December 1989; K.E. Batcher, Design of a Massively Parallel Processor, IEEE Transactions on Computers, Vol. C-29 No. 9, pp. 836-840

September 1980; and L. Uhr, <u>Multi-Computer Architectures for Artificial Intelligence</u>, New York, N.Y., John Wiley & Sons, Ch. 8, p. 97, 1987.

In the nearest neighbor torus connected computer of Fig. 1A multiple processing elements (PEs) are connected to their north, south, east and west neighbor PEs through torus connection paths MP and all PEs are operated in a synchronous single instruction multiple data (SIMD) fashion. Since a torus connected computer may be obtained by adding wraparound connections to a mesh-connected computer, a mesh-connected computer, one without wraparound connections, may be thought of as a subset of torus connected computers. As illustrated in Fig. 1B, each path MP may include T transmit wires and R receive wires, or as illustrated in Fig. 1C, each path MP may include B bidirectional wires. Although unidirectional and bidirectional communications are both contemplated by the invention, the total number of bus wires, excluding control signals, in a path will generally be referred to as k wires hereinafter, where k=B in a bidirectional bus design and k=T+R in a unidirectional bus design. It is assumed that a PE can transmit data to any of its neighboring PEs, but only one at a time. For example, each PE can transmit data to its east neighbor in one communication cycle. It is also assumed that a broadcast mechanism is present such that data and instructions can be dispatched from a controller simultaneously to all PEs in one broadcast dispatch period.

Although bit-serial inter-PE communications are typically employed to minimize wiring complexity, the wiring complexity of a torus-connected array nevertheless presents implementation problems. The conventional torus-connected array of Fig. 1A includes sixteen processing elements connected in a four by four array 10 of PEs. Each processing element $PE_{i,j}$ is labeled with its row and column number i and j, respectively. Each PE communicates to its nearest North (N), South (S), East (E) and West (W) neighbor with point to point connections. For example, the connection between $PE_{0,0}$

and $PE_{3,0}$ shown in Fig. 1A is a wraparound connection between $PE_{0,0}$'s N interface and $PE_{3,0}$'s south interface, representing one of the wraparound interfaces that forms the array into a torus configuration. In such a configuration, each row contains a set of N interconnections and, with N rows, there are N^2 horizontal connections. Similarly, with N columns having N vertical interconnections each, there are N^2 vertical interconnections. For the example of Fig. 1A, N=4. The total number of wires, such as the metallization lines in an integrated circuit implementation in an N x N torus-connected computer including wraparound connections, is therefore $2kN^2$, where k is the number of wires in each interconnection. The number k may be equal to one in a bit serial interconnection. For example with k=1 for the 4 x 4 array 10 as shown in Fig. 1A, $2kN^2=32$.

For a number of applications where N is relatively small, it is preferable that the entire PE array is incorporated in a single integrated circuit. The invention does not preclude implementations where each PE can be a separate microprocessor chip, for example. Since the total number of wires in a torus connected computer can be significant, the interconnections may consume a great deal of valuable integrated circuit ''real estate'', or the area of the chip taken up. Additionally, the PE interconnection paths quite frequently cross over one another complicating the IC layout process and possibly introducing noise to the communications lines through crosstalk. Furthermore, the length of wraparound links, which connect PEs at the North and South and at the East and West extremes of the array, increase with increasing array size. This increased length increases each communication line's capacitance, thereby reducing the line's maximum bit rate and introducing additional noise to the line.

Another disadvantage of the torus array arises in the context of transpose operations. Since a processing element and its transpose are separated by one or more intervening processing elements in the communications path, latency is

introduced in operations which employ transposes. For example, should the $PE_{2,1}$ require data from its transpose, $PE_{1,2}$, the data must travel through the intervening $PE_{1,1}$ or $PE_{2,2}$. Naturally, this introduces a delay into the operation, even if $PE_{1,1}$ and $PE_{2,2}$ are not otherwise occupied. However, in the general case where the PEs are implemented as micro-processor elements, there is a very good probability that $PE_{1,1}$ and $PE_{2,2}$ will be performing other operations and, in order to transfer data or commands from $PE_{1,2}$ to $PE_{2,1}$, they will have to set aside these operations in an orderly fashion. Therefore, it may take several operations to even begin transferring the data or commands from $PE_{1,2}$ to $PE_{1,1}$ and the operations $PE_{1,1}$ was forced to set aside to transfer the transpose data will also be delayed. Such delays snowball with every intervening PE and significant latency is introduced for the most distant of the transpose pairs. For example the $PE_{3,1}/PE_{1,3}$ transpose pair of Fig. 1A, has a minimum of three intervening PEs, requiring a latency of four communication steps and could additionally incur the latency of all the tasks which must be set aside in all those PEs in order to transfer data between $PE_{3,1}$ and $PE_{1,3}$ in the general case.

Recognizing such limitations of torus connected arrays, new approaches to arrays have been disclosed in U.S. Patent No. 5,612,908; A Massively Parallel Diagonal Fold Array Processor, G.G. Pechanek et al., 1993 International Conference on Application Specific Array Processors, pp. 140-143, October 25-27, 1993, Venice, Italy, and Multiple Fold Clustered Processor Torus Array, G.G. Pechanek, et. al., Proceedings Fifth NASA Symposium on VLSI Design, pp. 8.4.1-11, November 4-5, 1993, University of New Mexico, Albuquerque, New Mexico which are incorporated by reference herein in their entirety. The operative technique of these torus array organizations is the folding of arrays of PEs using the diagonal PEs of the conventional nearest neighbor torus as the foldover edge. As illustrated in the array 20 of Fig. 2, these techniques may be employed to substantially reduce inter-PE wiring, to reduce

the number and length of wraparound connections, and to
position PEs in close proximity to their transpose PEs. This
processor array architecture is disclosed, by way of example,
in U.S. Patent Nos. 5,577,262, 5,612,908, and EP 0,726,532 and

5   EP 0,726,529 which were invented by the same inventor as the
present invention and is incorporated herein by reference in
its entirety. While such arrays provide substantial benefits
over the conventional torus architecture, due to the
irregularity of PE combinations, for example in a single fold

10   diagonal fold mesh, some PEs are clustered ``in twos'', others
are single, in a three fold diagonal fold mesh there are
clusters of four PEs and eight PEs. Due to an overall
triangular shape of the arrays, the diagonal fold type of
array presents substantial obstacles to efficient, inexpensive

15   integrated circuit implementation. Additionally, in a
diagonal fold mesh as in EP 0,726,532 and EP 0,726,529, and
other conventional mesh architectures, the interconnection
topology is inherently part of the PE definition. This fixes
the PE's position in the topology, consequently limiting the

20   topology of the PEs and their connectivity to the fixed
configuration that is implemented. Thus, a need exists for
further improvements in processor array architecture and
processor interconnection.

<div align="center">SUMMARY OF THE INVENTION</div>

25   The present invention is directed to an array of
processing elements which substantially reduce the array's
interconnection wiring requirements when compared to the
wiring requirements of conventional torus processing element
arrays. In a preferred embodiment, one array in accordance

30   with the present invention achieves a substantial reduction in
the latency of transpose operations. Additionally, the
inventive array decouples the length of wraparound wiring from
the array's overall dimensions, thereby reducing the length of
the longest interconnection wires. Also, for array

35   communication patterns that cause no conflict between the
communicating PEs, only one transmit port and one receive port

are required per PE, independent of the number of neighborhood
connections a particular topology may require of its PE nodes.
A preferred integrated circuit implementation of the array
includes a combination of similar processing element clusters

5   combined to present a rectangular or square outline. The
similarity of processing elements, the similarity of
processing element clusters, and the regularity of the array's
overall outline make the array particularly suitable for
cost-effective integrated circuit manufacturing.

10   To form an array in accordance with the present
invention, processing elements may first be combined into
clusters which capitalize on the communications requirements
of single instruction multiple data ("SIMD") operations.
Processing elements may then be grouped so that the elements

15   of one cluster communicate within a cluster and with members
of only two other clusters. Furthermore, each cluster's
constituent processing elements communicate in only two
mutually exclusive directions with the processing elements of
each of the other clusters. By definition, in a SIMD torus

20   with unidirectional communication capability, the North/South
directions are mutually exclusive with the East/West
directions. Processing element clusters are, as the name
implies, groups of processors formed preferably in close
physical proximity to one another. In an integrated circuit

25   implementation, for example, the processing elements of a
cluster preferably would be laid out as close to one another
as possible, and preferably closer to one another than to any
other processing element in the array. For example, an array
corresponding to a conventional four by four torus array of

30   processing elements may include four clusters of four elements
each, with each cluster communicating only to the North and
East with one other cluster and to the South and West with
another cluster, or to the South and East with one other
cluster and to the North and West with another cluster. By

35   clustering PEs in this manner, communications paths between PE
clusters may be shared, through multiplexing, thus

substantially reducing the interconnection wiring required for the array.

In a preferred embodiment, the PEs comprising a cluster are chosen so that processing elements and their transposes are located in the same cluster and communicate with one another through intra-cluster communications paths, thereby eliminating the latency associated with transpose operations carried out on conventional torus arrays. Additionally, since the conventional wraparound path is treated the same as any PE-to-PE path, the longest communications path may be as short as the inter-cluster spacing, regardless of the array's overall dimension. According to the invention an N x M torus may be transformed into an array of M clusters of N PEs, or into N clusters of M PEs.

These and other features, aspects and advantages of the invention will be apparent to those skilled in the art from the following detailed description, taken together with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1A is a block diagram of a conventional prior art 4 x 4 nearest neighbor connected torus processing element (PE) array;

Fig. 1B illustrates how the prior art torus connection paths of Fig. 1A may include T transmit and R receive wires;

Fig. 1C illustrates how prior art torus connection paths of Fig. 1A may include B bidirectional wires;

Fig. 2 is a block diagram of a prior art diagonal folded mesh;

Fig. 3A is a block diagram of a processing element which may suitably be employed within the PE array of the present invention;

Fig. 3B is a block diagram of an alternative processing element which may suitably be employed within the PE array of the present invention;

Fig. 4 is a tiling of a 4 x 4 torus which illustrates all the torus's inter-PE communications links;

Figs. 5A through 5G are tilings of a 4 x 4 torus which illustrate the selection of PEs for cluster groupings in accordance with the present invention;

Fig. 6 is a tiling of a 4 x 4 torus which illustrates alternative grouping of PEs for clusters;

Fig. 7 is a tiling of a 3 x 3 torus which illustrates the selection of PEs for PE clusters;

Fig. 8 is a tiling of a 3 x 5 torus which illustrates the selection of PEs for PE clusters;

Fig. 9 is a block diagram illustrating an alternative, rhombus/cylinder approach to selecting PEs for PE clusters;

Fig. 10 is a block diagram which illustrates the inter-cluster communications paths of the new PE clusters;

Figs. 11A and 11B illustrate alternative rhombus/cylinder approaches to PE cluster selection;

Fig. 12 is a block diagram illustration of the rhombus/cylinder PE selection process for a 5 x 4 PE array;

Fig. 13 is a block diagram illustration of the rhombus/cylinder PE selection process for a 4 x 5 PE array;

Fig. 14 is a block diagram illustration of the rhombus/cylinder PE selection process for a 5 x 5 PE array;

Figs. 15A through 15D are block diagram illustrations of inter-cluster communications paths for 3, 4, 5, and 6 cluster by 6 PE arrays, respectively;

Fig. 16 is a block diagram illustrating East/South communications paths within an array of four four-member clusters;

Fig. 17 is a block diagram illustration of East/South and West/North communications paths within an array of four four-member clusters;

Fig. 18 is a block diagram illustrating one of the clusters of the embodiment of Fig. 17, which illustrates in greater detail a cluster switch and its interface to the illustrated cluster;

Figs. 19A and 19B illustrate a convolution window and convolution path, respectively, employed in an exemplary

convolution which may advantageously be carried out on the new array processor of the present invention;

Figs. 19C and 19D are block diagrams which respectively illustrate a portion of an image within a 4 x 4 block and the block loaded into conventional torus locations; and

Figs. 20A through 24B are block diagrams which illustrate the state of a manifold array in accordance with the present invention at the end of each convolution operational step.

DETAILED DESCRIPTION

In one embodiment, a new array processor in accordance with the present invention combines PEs in clusters, or groups, such that the elements of one cluster communicate with members of only two other clusters and each cluster's constituent processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By clustering PEs in this manner, communications paths between PE clusters may be shared, thus substantially reducing the interconnection wiring required for the array. Additionally, each PE may have a single transmit port and a single receive port or, in the case of a bidirectional sequential or time sliced transmit/receive communication implementation, a single transmit/receive port. As a result, the individual PEs are decoupled from the topology of the array. That is, unlike a conventional torus connected array where each PE has four bidirectional communication ports, one for communication in each direction, PEs employed by the new array architecture need only have one port. In implementations which utilize a single transmit and a single receive port, all PEs in the array may simultaneously transmit and receive. In the conventional torus, this would require four transmit and four receive ports, a total of eight ports, per PE, while in the present invention, one transmit port and one receive port, a total of two ports, per PE are required.

In one presently preferred embodiment, the PEs comprising a cluster are chosen so that processing elements and their

transposes are located in the same cluster and communicate
with one another through intra-cluster communications paths.
For convenience of description, processing elements are
referred to as they would appear in a conventional torus

5   array, for example, processing element $PE_{0,0}$ is the processing
element that would appear in the "Northwest" corner of a
conventional torus array. Consequently, although the layout
of the new cluster array is substantially different from that
of a conventional array

10   processor, the same data would be supplied to corresponding
processing elements of the conventional torus and new cluster
arrays. For example, the $PE_{0,0}$ element of the new cluster
array would receive the same data to operate on as the $PE_{0,0}$
element of a conventional torus-connected array.

15   Additionally, the directions referred to in this description
will be in reference to the directions of a torus-connected
array. For example, when communications between processing
elements are said to take place from North to South, those
directions refer to the direction of communication within a

20   conventional torus-connected array.

        The PEs may be single microprocessor chips that may be of
a simple structure tailored for a specific application.
Though not limited to the following description, a basic PE
will be described to demonstrate the concepts involved. The

25   basic structure of a PE 30 illustrating one suitable
embodiment which may be utilized for each PE of the new PE
array of the present invention is illustrated in Fig. 3A. For
simplicity of illustration, interface logic and buffers are
not shown. A broadcast instruction bus 31 is connected to

30   receive dispatched instructions from a SIMD controller 29, and
a data bus 32 is connected to receive data from memory 33 or
another data source external to the PE 30. A register file
storage medium 34 provides source operand data to execution
units 36. An instruction decoder/controller 38 is connected

35   to receive instructions through the broadcast instruction bus
31 and to provide control signals 21 to registers within the

register file 34 which, in turn, provide their contents as
operands via path 22 to the execution units 36. The execution
units 36 receive control signals 23 from the instruction
decoder/controller 38 and provide results via path 24 to the
register file 34. The instruction decoder/controller 38 also
provides cluster switch enable signals on an output the line
39 labeled Switch Enable. The function of cluster switches
will be discussed in greater detail below in conjunction with
the discussion of Fig. 18. Inter-PE communications of data or
commands are received at receive input 37 labeled Receive and
are transmitted from a transmit output 35 labeled Send.

Fig. 3B shows an alternative PE representation 30' that
includes an interface control unit 50 which provides data
formatting operations based upon control signals 25 received
from the instruction decoder/controller 38. Data formatting
operations can include, for example, parallel to serial and
serial to parallel conversions, data encryption, and data
format conversions to meet various standards or interface
requirements.

A conventional 4 x 4 nearest neighbor torus of PEs of the
same type as the PE 30 illustrated in Fig. 3A is shown
surrounded by tilings of itself in Fig. 4. The center 4 x 4
torus 40 is encased by a ring 42 which includes the wraparound
connections of the torus. The tiling of Fig. 4 is a
descriptive aid used to ''flatten out'' the wraparound
connections and to thereby aid in explanation of the preferred
cluster forming process utilized in the array of one
embodiment of the present invention. For example, the
wraparound connection to the west from PE_{00}, is $PE_{0,3}$, that from
the $PE_{1,3}$ to the east is $PE_{1,0}$, etc., as illustrated within the
block 42. The utility of this view will be more apparent in
relation to the discussion below of Figs. 5A-5G.

In Fig. 5A, the basic 4 x 4 PE torus is once again
surrounded by tilings of itself. The present invention
recognizes that communications to the East and South from $PE_{0,0}$
involve $PE_{0,1}$ and $PE_{1,0}$, respectively. Furthermore, the PE

which communicates to the east to $PE_{1,0}$ is $PE_{1,3}$ and $PE_{1,3}$
communicates to the South to $PE_{2,3}$. Therefore, combining the
four PEs, $PE_{0,0}$, $PE_{1,3}$, $PE_{2,2}$, and $PE_{3,1}$ in one cluster yields a
cluster 44 from which PEs communicate only to the South and
East with another cluster 46 which includes PEs, $PE_{0,1}$, $PE_{1,0}$,
$PE_{2,3}$ and $PE_{3,2}$. Similarly, the PEs of cluster 46 communicate
to the South and East with the PEs of cluster 48 which
includes PEs, $PE_{0,2}$, $PE_{1,1}$, $PE_{2,0}$, and $PE_{3,3}$. The PEs, $PE_{0,3}$, $PE_{1,2}$,
$PE_{2,1}$, and $PE_{3,0}$ of cluster 50 communicate to the South and East
with cluster 44. This combination yields clusters of PEs
which communicate with PEs in only two other clusters and
which communicate in mutually exclusive directions to those
clusters. That is, for example, the PEs of cluster 48
communicate only to the South and East with the PEs of cluster
50 and only to the North and West with the PEs of cluster 46.
   It is this exemplary of grouping of PEs which permits the
inter-PE connections within an array in accordance with the
present invention to be substantially reduced in comparison
with the requirements of the conventional nearest neighbor
torus array.
      Many other combinations are possible. For example,
starting again with $PE_{0,0}$ and grouping PEs in relation to
communications to the North and East yields clusters 52, 54,
56 and 58 of Fig 5B. These clusters may be combined in a way
which greatly reduces the interconnection requirements of the
PE array and which reduces the length of the longest inter-PE
connection. However, these clusters do not combine PEs and
their transposes as the clusters 44-50 in Fig. 5A do. That
is, although transpose pairs $PE_{0,2}/PE_{2,0}$ and $PE_{1,3}/PE_{3,1}$ are
contained in cluster 56, the transpose pair $PE_{0,1}/PE_{1,0}$ is split
between clusters 54 and 58. An array in accordance with the
presently preferred embodiment employs only clusters such as
44-50 which combine all PEs with their transposes within
clusters. For example, in Fig. 5A the $PE_{3,1}/PE_{1,3}$ transpose
pair is contained within cluster 44, the $PE_{3,2}$, $PE_{2,3}$ and
$PE_{1,0}/PE_{0,1}$ transpose pairs are contained within cluster 46, the

$PE_{0,2}/PE_{2,0}$ transpose pair is contained within cluster 48, and the $PE_{3,0}/PE_{0,3}$ and $PE_{2,1}/PE_{1,2}$ transpose pairs are contained within cluster 50. Clusters 60, 62, 64 and 68 of Fig 5C are formed, starting at $PE_{0,0}$, by combining PEs which communicate to the North and West. Note that cluster 60 is equivalent to cluster 44, cluster 62 is equivalent to cluster 46, cluster 64 is equivalent to cluster 48 and cluster 68 is equivalent to cluster 50. Similarly, clusters 70 through 76 of Fig. 5D, formed by combining PEs which communicate to the South and West, are equivalent to clusters 52 through 58, respectively of Fig. 5B. As demonstrated in Fig. 5E, clusters 45, 47, 49 and 51, which are equivalent to the preferred clusters 48, 50, 44 and 46 may be obtained from any ''starting point'' within the torus 40 by combining PEs which communicate to the South and East.

Another clustering is depicted in Fig. 5F where clusters 61, 63, 65, and 67 form a criss cross pattern in the tilings of the torus 40. This clustering demonstrates that there are a number of ways in which to group PEs to yield clusters which communicate with two other clusters in mutually exclusive directions. That is, $PE_{0,0}$ and $PE_{2,2}$ of cluster 65 communicate to the East with $PE_{0,1}$ and $PE_{2,3}$, respectively, of cluster 61. Additionally, $PE_{1,1}$ and $PE_{3,3}$ of cluster 65 communicate to the West with $PE_{1,0}$ and $PE_{3,2}$, respectively, of cluster 61. As will be described in greater detail below, the Easterly communications paths just described, that is, those between $PE_{0,0}$ and $PE_{0,1}$ and between $PE_{2,2}$ and $PE_{2,3}$, and other inter-cluster paths may be combined with mutually exclusive inter-cluster communications paths, through multiplexing for example, to reduce by half the number of interconnection wires required for inter-PE communications. The clustering of Fig. 5F also groups transpose elements within clusters.

One aspect of the new array's scalability is demonstrated by Fig. 5G, where a 4 X 8 torus array is depicted as two 4 X 4 arrays 40A and 40B. One could use the techniques described to this point to produce eight four-PE clusters from a 4 X 8

torus array. In addition, by dividing the 4 X 8 torus into
two 4 X 4 toruses and combining respective clusters into
clusters, that is clusters 44A and 44B, 46A and 46B, and so
on, for example, four eight-PE clusters with all the

5   connectivity and transpose relationships of the 4 X 4
subclusters contained in the eight four-PE cluster
configuration is obtained. This cluster combining approach is
general and other scalings are possible.

       The presently preferred, but not sole, clustering process

10   may also be described as follows. Given an N X N basic torus
$PE_{i,j}$, where i = 0,1,2, . . . N-1 and j = 0, 1, 2, . . . N-1,
the preferred, South- and East-communicating clusters may be
formed by grouping $PE_{i,j}$, $PE_{(i+1)(ModN), (j+N-1)(ModN)}$, $PE_{(i+2)(ModN), (j+N-2)(ModN)}$,
. . . , $PE_{(i+N-1)(ModN), (j+N-(N-1))(ModN)}$. This formula can be rewritten

15   for an N X N torus array with N clusters of N PEs in which the
cluster groupings can be formed by selecting an i and a j, and
then using the formula: $PE_{(i+a)(ModN),(j+N-a)(ModN)}$ for any i,j and for
all a $\in \{0,1,...,N-1\}$.

       Fig. 6 illustrates the production of clusters 44 through

20   50 beginning with $PE_{1,3}$ and combining PEs which communicate to
the South and East. In fact, the clusters 44 through 50,
which are the clusters of the preferred embodiment of a 4 x 4
torus equivalent of the new array, are obtained by combining
South and East communicating PEs, regardless of what PE within

25   the basic N X N torus 40 is used as a starting point. Figs. 7
and 8 illustrate additional examples of the approach, using 3
x 3 and 3 x 5 toruses, respectively.

       Another, equivalent way of viewing the cluster-building
process is illustrated in Fig. 9. In this and similar figures

30   that follow, wraparound wires are omitted from the figure for
the sake of clarity. A conventional 4 x 4 torus is first
twisted into a rhombus, as illustrated by the leftward shift
of each row. This shift serves to group transpose PEs in
``vertical slices'' of the rhombus. To produce equal-size

35   clusters the rhombus is, basically, formed into a cylinder.
That is, the left-most, or western-most, vertical slice 80 is

wrapped around to abut the eastern-most $PE_{0,3}$ in its row. The
vertical slice 82 to the east of slice 80 is wrapped around to
abut $PE_{0,0}$ and $PE_{1,3}$, and the next eastward vertical slice 84 is
wrapped around to abut $PE_{0,1}$, $PE_{1,0}$ and $PE_{2,3}$. Although, for the

5   sake of clarity, all connections are not shown, all
connections remain the same as in the original 4 x 4 torus.
The resulting vertical slices produce the clusters of the
preferred embodiment 44 through 50 shown in Fig. 5A, the same
clusters produced in the manner illustrated in the discussion

10   related to Figs. 5A and 6. In Fig. 10, the clusters created
in the rhombus/cylinder process of Fig. 9 are ''peeled open''
for illustrative purposes to reveal the inter-cluster
connections. For example, all inter-PE connections from
cluster 44 to cluster 46 are to the South and East, as are

15   those from cluster 46 to cluster 48 and from cluster 48 to
cluster 50 and from cluster 50 to cluster 44. This
commonality of inter-cluster communications, in combination
with the nature of inter-PE communications in a SIMD process
permits a significant reduction in the number of inter-PE

20   connections. As discussed in greater detail in relation to
Figs. 16 and 17 below, mutually exclusive communications,
e.g., communications to the South and East from cluster 44 to
cluster 46 may be multiplexed onto a common set of
interconnection wires running between the clusters.

25   Consequently, the inter-PE connection wiring of the new array,
hereinafter referred to as the ''manifold array'', may be
substantially reduced, to one half the number of
interconnection wires associated with a conventional nearest
neighbor torus array.

30   The cluster formation process used to produce a manifold
array is symmetrical and the clusters formed by taking
horizontal slices of a vertically shifted torus are the same
as clusters formed by taking vertical slices of a horizontally
shifted torus. Figs. 11A and 11B illustrate the fact that the

35   rhombus/cylinder technique may also be employed to produce the
preferred clusters from horizontal slices of a vertically

shifted torus. In Fig. 11A the columns of a conventional 4 x
4 torus array are shifted vertically to produce a rhombus and
in Fig 11B the rhombus is wrapped into a cylinder. Horizontal
slices of the resulting cylinder provide the preferred

5   clusters 44 through 50. Any of the techniques illustrated to
this point may be employed to create clusters for manifold
arrays which provide inter-PE connectivity equivalent to that
of a conventional torus array, with substantially reduced
inter-PE wiring requirements.

10   As noted in the summary, the above clustering process is
general and may be employed to produce manifold arrays of M
clusters containing N PEs each from an N x M torus array. For
example, the rhombus/cylinder approach to creating four
clusters of five PEs, for a 5 x 4 torus array equivalent is

15   illustrated in Fig. 12. Note that the vertical slices which
form the new PE clusters, for example, $PE_{4,0}$, $PE_{3,1}$, $PE_{2,2}$, $PE_{1,3}$,
and $PE_{0,0}$ maintain the transpose clustering relationship of the
previously illustrated 4 x 4 array. Similarly, as illustrated
in the diagram of Fig. 13, a 4 x 5 torus will yield five

20   clusters of four PEs each with the transpose relationship only
slightly modified from that obtained with a 4 x 4 torus. In
fact, transpose PEs are still clustered together, only in a
slightly different arrangement than with the 4 x 4 clustered
array. For example, transpose pairs $PE_{1,0}/PE_{0,1}$ and $PE_{2,3}/PE_{3,2}$

25   were grouped in the same cluster within the preferred 4 x 4
manifold array, but they appear, still paired, but in separate
clusters in the 4 x 5 manifold array of Fig. 13. As
illustrated in the cluster-selection diagram of Fig. 14, the
diagonal PEs, $PE_{i,j}$ where i = j, in an odd number by odd number

30   array are distributed one per, cluster.
       The block diagrams of Figs. 15A-15D illustrate the
inter-cluster connections of the new manifold array. To
simplify the description, in the following discussion,
unidirectional connection paths are assumed unless otherwise

35   stated. Although, for the sake of clarity, the invention is
described with parallel interconnection paths, or buses,

represented by individual lines. Bit-serial communications,
in other words buses having a single line, are also
contemplated by the invention. Where bus multiplexers or bus
switches are used, the multiplexer and/or switches are
5   replicated for the number of lines in the bus. Additionally,
with appropriate network connections and microprocessor chip
implementations of PEs, the new array may be employed with
systems which allow dynamic switching between MIMD, SIMD and
SISD modes, as described in US Patent 5,475,856 to P.M. Kogge,
10   entitled, Dynamic Multi-Mode Parallel Processor Array
Architecture, which is hereby incorporated by reference.

    In Fig. 15A, clusters 80, 82 and 84 are three PE clusters
connected through cluster switches 86 and inter-cluster links
88 to one another. To understand how the manifold array PEs
15   connect to one another to create a particular topology, the
connection view from a PE must be changed from that of a
single PE to that of the PE as a member of a cluster of PEs.
For a manifold array operating in a SIMD unidirectional
communication environment, any PE requires only one transmit
20   port and one receive port, independent of the number of
connections between the PE and any of its directly attached
neighborhood of PEs in the conventional torus. In general,
for array communication patterns that cause no conflicts
between communicating PEs, only one transmit and one receive
25   port are required per PE, independent of the number of
neighborhood connections a particular topology may require of
its PEs.

    Four clusters, 44 through 50, of four PEs each are
combined in the array of Fig. 15B. Cluster switches 86 and
30   communication paths 88 connect the clusters in a manner
explained in greater detail in the discussion of Figs. 16, 17,
and 18 below. Similarly, five clusters, 90 through 98, of
five PEs each are combined in the array of Fig. 15C. In
practice, the clusters 90-98 are placed as appropriate to ease
35   integrated circuit layout and to reduce the length of the
longest inter-cluster connection. Fig. 15D illustrates a

manifold array of six clusters, 99, 100, 101, 102, 104, and
106, having six PEs each. Since communication paths 86 in the
new manifold array are between clusters, the wraparound
connection problem of the conventional torus array is

5   eliminated. That is, no matter how large the array becomes,
no interconnection path need be longer than the basic
inter-cluster spacing illustrated by the connection paths 88.
This is in contrast to wraparound connections of conventional
torus arrays which must span the entire array.

10   The block diagram of Fig. 16 illustrates in greater
detail a preferred embodiment of a four cluster, sixteen PE,
manifold array. The clusters 44 through 50 are arranged, much
as they would be in an integrated circuit layout, in a
rectangle or square. The connection paths 88 and cluster

15   switches are illustrated in greater detail in this figure.
Connections to the South and East are multiplexed through the
cluster switches 86 in order to reduce the number of
connection lines between PEs. For example, the South
connection between $PE_{1,2}$ and $PE_{2,2}$ is carried over a connection

20   path 110, as is the East connection from $PE_{2,1}$ to $PE_{2,2}$. As
noted above, each connection path, such as the connection path
110 may be a bit-serial path and, consequently, may be
effected in an integrated circuit implementation by a single
metallization line. Additionally, the connection paths are

25   only enabled when the respective control line is asserted.
These control lines can be generated by the instruction
decoder/controller 38 of each $PE_{3,0}$, illustrated in Fig. 3A.
Alternatively, these control lines can be generated by an
independent instruction decoder/controller that is included in

30   each cluster switch. Since there are multiple PEs per switch,
the multiple enable signals generated by each PE are compared
to make sure they have the same value in order to ensure that
no error has occurred and that all PEs are operating
synchronously. That is, there is a control line associated

35   with each noted direction path, N for North, S for South, E
for East, and W for West. The signals on these lines enable

the multiplexer to pass data on the associated data path through the multiplexer to the connected PE. When the control signals are not asserted the associated data paths are not enabled and data is not transferred along those paths through
5 the multiplexer.

The block diagram of Fig. 17 illustrates in greater detail the interconnection paths 88 and switch clusters 86 which link the four clusters 44 through 50. In this figure, the West and North connections are added to the East and South
10 connections illustrated in Fig. 16. Although, in this view, each processing element appears to have two input and two output ports, in the preferred embodiment another layer of multiplexing within the cluster switches brings the number of communications ports for each PE down to one for input and one
15 for output. In a standard torus with four neighborhood transmit connections per PE and with unidirectional communications, that is, only one transmit direction enabled per PE, there are four multiplexer or gated circuit transmit paths required in each PE. A gated circuit may suitably
20 include multiplexers, AND gates, tristate driver/receivers with enable and disable control signals, and other such interface enabling/disabling circuitry. This is due to the interconnection topology defined as part of the PE. The net result is that there are $4N^2$ multiple transmit paths in the
25 standard torus. In the manifold array, with equivalent connectivity and unlimited communications, only $2N^2$ multiplexed or gated circuit transmit paths are required. This reduction of $2N^2$ transmit paths translates into a significant savings in integrated circuit real estate area, as
30 the area consumed by the multiplexers and $2N^2$ transmit paths is significantly less than that consumed by $4N^2$ transmit paths.

A complete cluster switch 86 is illustrated in greater detail in the block diagram of Fig. 18. The North, South,
35 East, and West outputs are as previously illustrated. Another layer of multiplexing 112 has been added to the cluster switch

86. This layer of multiplexing selects between East/South
reception, labeled A, and North/West reception, labeled B,
thereby reducing the communications port requirements of each
PE to one receive port and one send port. Additionally,
5   multiplexed connections between transpose PEs, $PE_{1,3}$ and $PE_{3,1}$,
are effected through the intra-cluster transpose connections
labeled T. When the T multiplexer enable signal for a
particular multiplexer is asserted, communications from a
transpose PE are received at the PE associated with the
10 multiplexer. In the preferred embodiment, all clusters
include transpose paths such as this between a PE and its
transpose PE. These figures illustrate the overall connection
scheme and are not intended to illustrate how a multi-layer
integrated circuit implementation may accomplish the entirety
15 of the routine array interconnections that would typically be
made as a routine matter of design choice. As with any
integrated circuit layout, the IC designer would analyze
various tradeoffs in the process of laying out an actual IC
implementation of an array in accordance with the present
20 invention. For example, the cluster switch may be distributed
within the PE cluster to reduce the wiring lengths of the
numerous interfaces.

To demonstrate the equivalence to a torus array's
communication capabilities and the ability to execute an image
25 processing algorithm on the Manifold Array, a simple 2D
convolution using a 3 x 3 window, Fig. 19A, will be described
below. The Lee and Aggarwal algorithm for convolution on a
torus machine will be used. See, S.Y. Lee and J. K. Aggarwal,
Parallel 2D Convolution on a Mesh Connected Array Processor,
30 IEEE Transactions on Patter Analysis and Machine Intelligence,
Vol. PAMI-9, No. 4, pp. 590-594, July 1987. The internal
structure of a basic PE 30, Fig. 3A, is used to demonstrate
the convolution as executed on a 4 x 4 Manifold Array with 16
of these PEs. For purposes of this example, the Instruction
35 Decoder/Controller also provides the Cluster Switch
multiplexer Enable signals. Since there are multiple PEs per

switch, the multiple enable signals are compared to be equal to ensure no error has occurred and all PEs are operating in synchronism.

Based upon the S.Y. Lee and J. K. Aggarwal algorithm for convolution, the Manifold array would desirably be the size of the image, for example, an N x N array for a N x N image. Due to implementation issues it must be assumed that the array is smaller than N x N for large N. Assuming the array size is C x C, the image processing can be partitioned into multiple C x C blocks, taking into account the image block overlap required by the convolution window size. Various techniques can be used to handle the edge effects of the N x N image. For example, pixel replication can be used that effectively generates an (N+1) x (N+1) array. It is noted that due to the simplicity of the processing required, a very small PE could be defined in an application specific implementation. Consequently, a large number of PEs could be placed in a Manifold Array organization on a chip thereby improving the efficiency of the convolution calculations for large image sizes.

The convolution algorithm provides a simple means to demonstrate the functional equivalence of the Manifold Array organization to a torus array for North/East/South/West nearest neighbor communication operations. Consequently, the example focuses on the communications aspects of the algorithm and, for simplicity of discussion, a very small 4 x 4 image size is used on a 4 x 4 Manifold array. Larger N x N images can be handled in this approach by loading a new 4 x 4 image segment into the array after each previous 4 x 4 block is finished. For the 4 x 4 array no wrap around is used and for the edge PEs 0's are received from the virtual PEs not present in the physical implementation. The processing for one 4 x 4 block of pixels will be covered in this operating example.

To begin the convolution example, it is assumed that the PEs have already been initialized by a SIMD controller, such as controller 29 of Fig. 3A, and the initial 4 x 4 block of

pixels has been loaded through the data bus to register R1 in each PE, in other words, one pixel per PE has been loaded. Fig. 19C shows a portion of an image with a 4 x 4 block to be loaded into the array. Fig. 19D shows this block loaded in the 4 x 4 torus logical positions. In addition, it is assumed that the accumulating sum register R0 in each PE has been initialized to zero. Though inconsequential to this algorithm, R2 has also been shown as initialized to zero. The convolution window elements are broadcast one at a time in each step of the algorithm. These window elements are received into register R2. The initial state of the machine prior to broadcasting the window elements is shown in Fig. 20A. The steps to calculate the sum of the weighted pixel values in a 3 x 3 neighborhood for all PEs follows.

The algorithm begins with the transmission (broadcasting) of the first window element W00 to all PEs. Once this is received in each PE, the PEs calculate the first $R0=R0+R2*R1$ or $R0=R0+W*P$. The result of the calculation is then communicated to a nearest neighbor PE according to the convolution path chosen, Fig. 19B. For simplicity of discussion it is assumed that each operational step to be described can be partitioned into three substeps each controlled by instructions dispatched from the controller: a broadcast window element step, a computation step, and a communications step. It is noted that improvements to this simplified approach can be developed, such as, beginning with major step 2, overlapping the window element broadcast step with the communications of result step. These points are not essential to the purpose of this description and would be recognized by one of ordinary skill in the art. A superscript is used to represent the summation step value as the operation proceeds. As an aid for following the communications of the calculated values, a subscript on a label indicates the source PE that the value was generated in. The convolution path for pixel {i,j} is shown in Fig. 19B. Figs. 20-24 indicate the state of the Manifold Array after each

computation step.

In Fig. 20B, W00 is broadcast to the PEs and each PE calculates $R0^1 = 0 + W00*R1$ and communicates $R0^1$ to the South PE where the received $R0^1$ value is stored in the PEs' register R0.

In Fig. 21A, W10 is broadcast to the PEs and each PE calculates $R0^2 = R0^1 + W10*R1$ and communicates $R0^2$ to the South PE where the received $R0^2$ value is stored in the PEs' register R0.

In Fig. 21B, W20 is broadcast to the PEs and each PE calculates $R0^3 = R0^2 + W20*R1$ and communicates $R0^3$ to the East PE where the received $R0^3$ value is stored in the PEs' register R0.

In Fig. 22A, W21 is broadcast to the PEs and each PE calculates $R0^4 = R0^3 + W21*R1$ and communicates $R0^4$ to the East PE where the received $R0^4$ value is stored in the PEs' register R0.

In Fig. 22B, W22 is broadcast to the PEs and each PE calculates $R0^5 = R0^4 + W22*R1$ and communicates $R0^5$ to the North PE where the received $R0^5$ value is stored in the PEs' register R0.

In Fig. 23A, W12 is broadcast to the PEs and each PE calculates $R0^6 = R0^5 + W12*R1$ and communicates $R0^6$ to the North PE where the received $R0^6$ value is stored in the PEs' register R0

In Fig. 23B, W02 is broadcast to the PEs and each PE calculates $R0^7 = R0^6 + W02*R1$ and communicates $R0^7$ to the West PE where the received R07 value is stored in the PEs' register R0

In Fig. 24A, W01 is broadcast to the PEs and each PE calculates $R0^8 = R0^7 + W01*R1$ and communicates $R0^8$ to the South PE where the received $R0^8$ value is stored in the PEs' register R0

In Fig. 24B, W11 is broadcast to the PEs and each PE calculates $R0^9 = R0^8 + W11*R1$ and End.

At the end of the above nine steps each $PE_{i,j}$ contains (with reference to Figure 19B):

$C_{i,j} = W00P_{i-1,j-1} + W10P_{i,j-1} + W20P_{i+1,j-1} + W21P_{i+1,j} + W22P_{i+1,j+1} + W12P_{i,j+1} + W02P_{i-1,j+1} + W01P_{i-1,j} + W11P_{i,j}.$

For example, for i=5, and j=6 $C_{5,6}$ = W00P4,5+W10P5,5
+W20P6,5+W21P6,6+W22P6,7+W12P5,7+W02P4,7+W01P4,6+W11P5,6.

It is noted that at the completion of this example, given
the operating assumptions, four valid convolution values have
been calculated, namely the ones in PEs {(1,1), (1,2), (2,1),
(2,2)}. This is due to the edge effects as discussed
previously. Due to the simple nature of the PE needed for
this algorithm, a large number of PEs can be incorporated on a
chip, thereby greatly increasing the efficiency of the
convolution calculation for large image sizes.

The above example demonstrates that the Manifold Array is
equivalent in its communications capabilities for the four -
North, East, South, and West - communications directions of a
standard torus while requiring only half the wiring expense of
the standard torus. Given the Manifold Array's capability to
communicate between transpose PEs, implemented with a regular
connection pattern, minimum wire length, and minimum cost, the
Manifold Array provides additional capabilities beyond the
standard torus. Since the Manifold Array organization is more
regular as it is made up of the same size clusters of PEs
while still providing the communications capabilities of
transpose and neighborhood communications, it represents a
superior design to the standard and diagonal fold toruses of
the prior art.

The foregoing description of specific embodiments of the
invention has been presented for the purposes of illustration
and description. It is not intended to be exhaustive or to
limit the invention to the precise forms disclosed, and many
modifications and variations are possible in light of the
above teachings. The embodiments were chosen and described in
order to best explain the principles of the invention and its
practical application, to thereby enable others skilled in the
art to best utilize the invention. It is intended that the
scope of the invention be limited only by the claims appended
hereto.